SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I2P121

Volume 13 Issue 2, 281-290, February 2026

Original Article

Cloud-Based IoT and AWS Architecture for Real-Time Cardiovascular Patient Monitoring

Dasaraju Chandra Mohan[1], R.Yogesh Rajkumar[2]

[1]*Department of CSE, Bharath Institute of Higher Education and Research, Selaiyur, Tambaram, Chennai, India.*
[2]*Department of Information Technology, Bharath Institute of Higher Education and Research, Selaiyur, Tambaram, Chennai, India.*

[1]*Corresponding Author : chandraraju9427@gmail.com*

Received: 24 November 2025 Revised: 26 December 2025 Accepted: 27 January 2026 Published: 20 February 2026

Abstract *- Cardiovascular disease requires continuous and timely monitoring to avert abrupt medical catastrophes, particularly for high-risk patients who cannot stay under continual hospital observation. The Internet of Things (IoT) has been the subject of much research into healthcare monitoring systems, but current solutions have several drawbacks, such as slow cloud updates, no real-time event-driven alerting, poor integration of scalable cloud services, and restricted access for multiple users. In this study, we provide an Internet of Things (IoT) architecture that runs on Amazon Web Services (AWS) to monitor heart patients in real-time. Through the use of wearable medical sensors coupled with an Internet of Things module based on the ESP8266, the system is able to gather physiological characteristics such as heart rate, temperature, and oxygen saturation. By using MQTT and TLS authentication, the collected data is safely sent to AWS IoT Core. It is then processed in real-time by AWS Lambda and saved in DynamoDB for scalable time-series management. Notifications of critical threshold breaches are sent automatically using AWS Simple Notification Service (SNS), allowing for quick action. In addition, two dashboards built on Android are created to provide medical professionals and guardians with access to real-time visualization and monitoring. When compared to traditional cloud-IoT healthcare systems, the suggested system outperforms them in terms of end-to-end latency, alert accuracy, and scalability. The results prove that the suggested event-based monitoring framework powered by AWS is an efficient, practical, and lightweight way to keep an eye on patients' heart health in real-time.*

Keywords *- Internet of Things, Cloud computing, AWS, Cardiovascular diseases.*

1. Introduction

The Internet of Things (IoT) has considerably enhanced healthcare by allowing continuous monitoring of patient vital signs via wearable sensors and wireless connectivity. [1]. Patients with cardiovascular disease, who are particularly vulnerable to life-threatening crises caused by rapid changes in vital signs, including heart rate, oxygen saturation, and temperature, must adhere to this strict protocol. [2].

Despite the abundance of IoT-based monitoring systems, the majority of them have issues such as slow scalability, insufficient integration of secure cloud services, and delayed cloud. While cloud platforms like AWS provide dependable and scalable infrastructure for healthcare IoT applications, the majority of current AWS-IoT deployments lack a comprehensive framework for real-time monitoring and emergency alerts tailored to healthcare. [3].

Consequently, a secure, event-driven, end-to-end cloud-IoT architecture is very necessary to guarantee quick clinical reaction and real-time cardiovascular monitoring. [4]. Medical care must be more accessible to improve patients' quality of life. Remote scheduling of medical equipment through IoT decreases medical equipment downtimes. Moreover, IoT systems provide accurate estimates of equipment replacement, thereby improving patient care, resource distribution, and utilization. Availability of efficient and effective communication channels enhances relations among the clinics, patients, and organizations. [5, 6] A comprehensive healthcare monitoring solution that incorporates clinical decision logic, inexpensive wearable sensor integration, workflows with multiple users, and patient-specific emergency alert handling is not provided by AWS's powerful cloud services like AWS IoT Core, Lambda, DynamoDB, SNS, and CloudWatch [7].

Lacking an end-to-end healthcare-oriented architecture, most current AWS IoT implementations primarily concentrate on generic device connectivity and data storage, rather than ensuring continuous physiological monitoring, real-time threshold-based clinical alerting, and user-friendly dashboards for doctors and guardians. [8].

In order to fill this void, the suggested system integrates wearable sensors with secure MQTT transmission, analytics driven by events in Lambda, storage in real-time

DynamoDB indexed by patient ID and timestamp, and automated SNS notifications for emergencies. [9]. Furthermore, the system offers a realistic deployment strategy that enhances alert response, decreases monitoring latency, allows many healthcare stakeholders to use the system at the same time, and offers continuous real-time viewing using Android apps. Consequently, what makes this AWS cloud platform unique is not the AWS services themselves, but rather the integrated healthcare-specific design and architecture that turns them into a platform for real-time patient monitoring and emergency response. [11]. The current state of cloud-based healthcare IoT monitoring systems is primarily concerned with storing and visualizing data, but they fall short when it comes to healthcare-specific workflow integration, scalable access for many users, and real-time event-driven clinical alerts. A comprehensive end-to-end framework for cardiovascular monitoring that includes medical threshold validation, continuous time-series indexing, and emergency reaction automation is not available on AWS, despite the fact that it offers IoT Core, Lambda, DynamoDB, and SNS. Consequently, this study fills that need by developing a secure, low-latency, AWS IoT infrastructure with a healthcare focus. This architecture will enable multi-stakeholder dashboards, automated alarm triggering, and real-time physiological streaming.

1.1. Important Achievements

An AWS IoT framework that monitors cardiovascular health using MQTT and TLS from beginning to end. Real-time analytics based on Lambda with automatic notifications for SNS emergencies. Timestamp and patient ID indexed scalable DynamoDB time series storage. Doctors and guardians may utilize multi-user monitoring dashboards. Statistical analysis based on measures for latency, error rate, packet delivery, and alert precision (Figure 1).

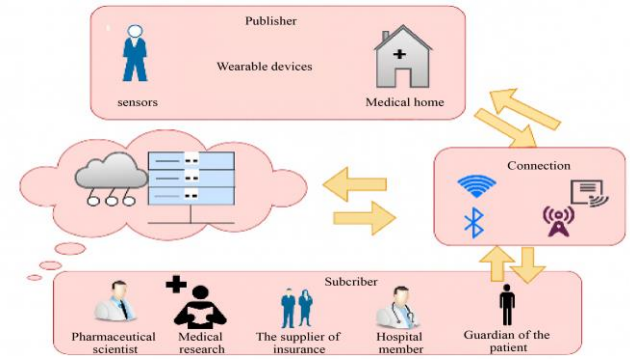

Fig. 1 Healthcare setup using conventional IoT and cloud computing

2. Related Work

A lot of the recent work in healthcare that makes use of the internet of things has been on creating architectures for remote patient monitoring and wearable sensor systems that can continuously measure vital signs like temperature, heart rate, and oxygen saturation (Research has shown that the Internet of Things (IoT) may help make healthcare more accessible by facilitating the remote capture of data in real-time and the making of clinical decisions, particularly in the areas of chronic illness management and care for the elderly.

Previous research has also highlighted the significance of combining cloud computing with the Internet of Things (IoT) to provide large-scale patient monitoring settings with effective data analytics, centralized access, and scalable storage. [14, 15]. There is a lot of talk in the literature about healthcare frameworks that use the cloud to handle and display data collected by sensors. Traditional methods of communication, such as HTTP and REST APIs, add unnecessary complexity and latency to certain systems, rendering them unfit for real-time heart rate monitoring. Because of its lightweight nature, MQTT has been suggested by other systems for communication. However, many of these frameworks still rely on planned synchronization or periodic batch uploads instead of continuous event-driven transmission.

Consequently, emergency response systems are less effective since alarm creation is often delayed. As an added complication, automated decision-making and real-time analytics are still missing from many of the current IoT healthcare systems. Several studies provide visual monitoring dashboards, but they do not include clinical threshold validation or intelligent warning systems that can inform medical professionals and caregivers immediately when anything out of the ordinary happens. Furthermore, scalability is still a big concern, as most of the designs that have been suggested have not been validated in large-scale settings and cannot handle concurrent user access and multi-patient monitoring. Since healthcare IoT devices deal with personal patient data, there has been a lot of talk about privacy and security concerns in the literature. Security in transmission, authentication via certificates, and access control techniques are all areas where many current solutions fall short, despite the fact that encryption and authentication are acknowledged as crucial needs. Because of this, they are susceptible to privacy breaches, data manipulation, and unwanted access.

Before developing a comprehensive architecture for real-time monitoring in healthcare, most research utilizing AWS concentrated on generic IoT connectivity and storage, even though AWS IoT Core, Lambda, DynamoDB, and SNS provide advanced capabilities for scalable deployment of IoT. Up until now, there has not been a comprehensive system that integrates clinically important features like indexed time-series storage, automatic emergency alerts, secure sensor streaming, event-driven processing, and multi-user monitoring dashboards. Consequently, there is a noticeable lack of research on a healthcare-specific cloud-IoT monitoring platform that is scalable, secure, low-latency, and capable of providing real-time cardiovascular patient surveillance and quick emergency reaction.

3. Methodology

3.1. System Architecture and Data Flow

The proposed framework integrates IoT sensors, cloud computing, and mobile applications for real-time patient health monitoring. Figure 2 illustrates the complete system architecture. Patient vital signs (pulse, temperature, heart

rate, oxygen saturation) are captured using medical-grade sensors integrated into a WeMos Board equipped with a Wi-Fi ESP8266 module. The collected data is encrypted using the AES algorithm and transmitted to AWS (Amazon Web Services) cloud infrastructure through secure API calls.

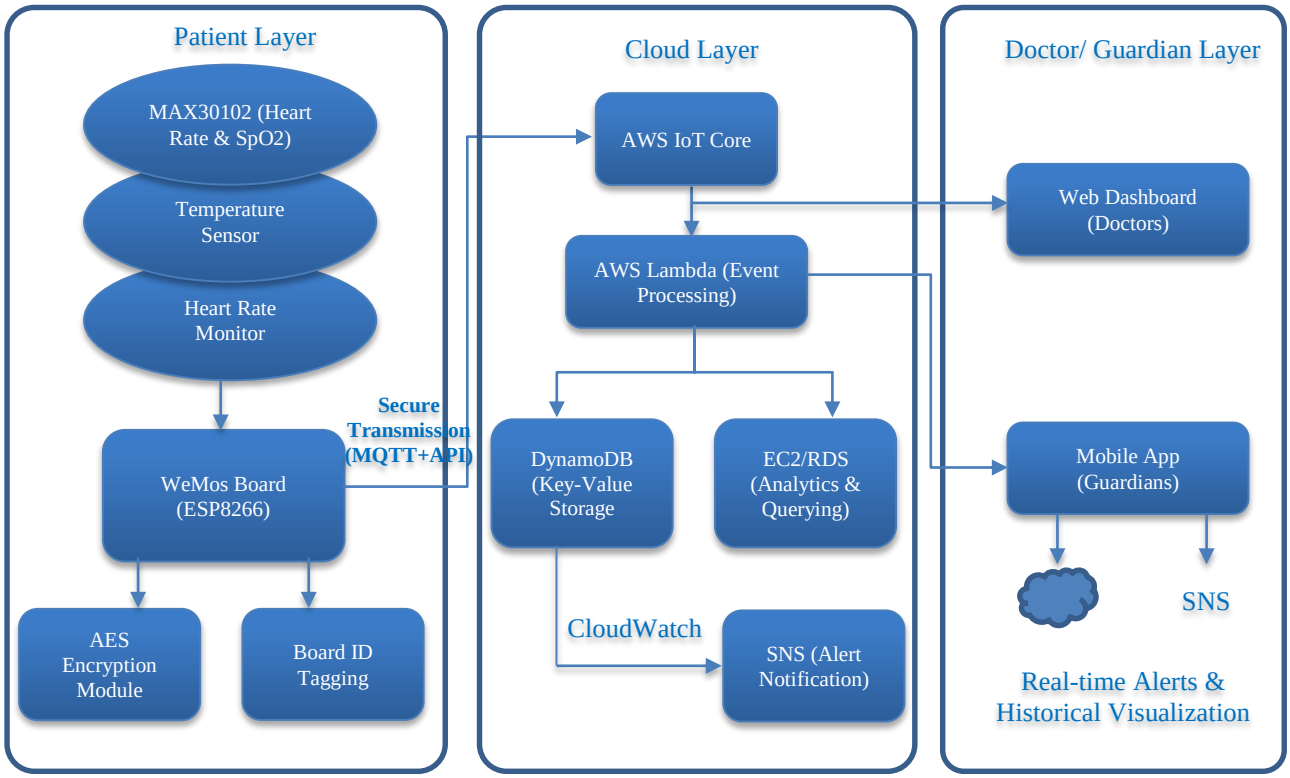

Fig. 2 System architecture

3.1.1. Cloud-based Data Acquisition and Processing Pipeline

To facilitate the collection, processing, and alarm generation of real-time data, the suggested cloud-based IoT monitoring system connects hardware and sensors with AWS cloud services. Physiological and environmental sensors are interfaced with the help of a Wi-Fi-enabled microcontroller unit (like the ESP8266) that operates at 3.3 V and has on-board flash memory. These sensors utilize readings, which are sampled at regular intervals and then sent via a private 2.4 GHz Wi-Fi network using the MQTT protocol; the parameters being monitored include temperature, heart rate, and oxygen saturation. Secure connection is ensured by publishing sensor data to AWS IoT Core using X.509 certificate-based authentication and TLS encryption. AWS Lambda functions validate, analyze, depending on thresholds, and preprocess the incoming data streams according to AWS IoT standards. The records are then stored in a DynamoDB database that is indexed by patient ID and timestamp. API Gateway services send notifications to the user interface when the critical thresholds are exceeded, thus raising alert situations. The steps in the entire flow of data that enable the ability to continuously monitor data, scale storage, and send timely alarms include sensors, a microcontroller, AWS IoT Core, Lambda, DynamoDB, and a mobile/web application. This architecture supports real-time health and industrial monitoring applications and ensures reliability, low latency, and scalability.

3.2. Data Collection Method

There are three main data collection sensors:

- MAX30102 Pulse Oximeter: Heart rate (bpm) and SpO2 (%). Included in the product range are heart rate monitors, pulse oximeters, and blood gas analyzers. Human Heart rate monitors, pulse oximeters, and blood gas analyzers are also built into the range of products.
- Digital Temperature Sensor: Measures the body temperature (°C) with accuracy of ±0.1-°C. Heart rate monitor: Records pulse rate in real time at a frequency of 1 Hz with a sample rate of 1 Hz. All sensor readings are timestamped and assigned the ID of a board that pertains to a specific patient. The sensor data are processed in the ESP8266 NodeMCU microcontroller and then securely transmitted to the cloud.

3.2.1. Ethical Approval

The physiological data used in this research were acquired using wearable Internet of Things (IoT) sensors in a controlled laboratory setting. Only anonymised Board IDs were used for patient association; no Personally Identifying Information (PII) was maintained. Before collecting sensor readings, participants were asked to provide their informed consent. The study conforms with research that adheres to ethical standards, as the evaluation was carried out as a prototype validation with participants' voluntary participation and anonymised data. Institutional Review Boards (IRBs) and formal Institutional Ethics Committees (IECs) will be needed for any future clinical deployment.

3.2.2. Description of Data Source

Wearable sensors based on the Internet of Things (IoT) that are fastened to the patient monitoring module provide the data used in this study. The device records the user's temperature in degrees Celsius, measures heart rate in Beats Per Minute (BPM), and oxygen saturation in percentages (SpO_2%) using a digital temperature sensor. The sensor values are recorded by the ESP8266-based WeMos/NodeMCU microcontroller, which is then timestamped and linked to a specific patient's board ID. Therefore, the dataset is created as a time-series stream of physiological characteristics, which are sampled at predetermined intervals and safely communicated to the AWS cloud. In doing so, we guarantee that the data we gather is reflective of actual monitoring settings and that it will be suitable for use in ongoing healthcare analytics hosted on the cloud.

3.3. Mathematical Framework

The proposed IoT-cloud healthcare monitoring system involves multiple sequential operations, including sensor data acquisition, secure transmission, cloud processing, database storage, and alert generation. These operations can be mathematically modeled as follows.

3.3.1. Data Acquisition Model

$$D_i(t) = f(s_i, t) \quad (1)$$

Where:

- $D_i(t)$= data acquired from the sensor $i$at time t
- s_i= sensor type (MAX30102, temperature sensor, etc.)
- $f(\cdot)$= sensor measurement function
- t= sampling time instance

3.3.2. Sampling Rate Model

$$t_k = t_0 + k\Delta t \quad (2)$$

Where:

- t_k= k^{th}sampling instant
- t_0= initial sampling time
- Δt= sampling interval (seconds)
- k= sample index ($k = 1,2,3,\ldots,N$)
- N= total number of samples

3.3.3. Data Transmission Time

$$T_{tx}(D_i) = \frac{|\ D_i\ |}{BW_{WiFi}} \quad (3)$$

Where:

- $T_{tx}(D_i)$= transmission time for sensor packet D_i
- $|\ D_i\ |$= size of data packet (bits/bytes)
- BW_{WiFi}= available Wi-Fi bandwidth (bps)

3.3.4. Total Communication Delay Model

$$T_{comm} = T_{tx} + T_{prop} + T_{queue} \quad (4)$$

Where:

- T_{comm}= total communication delay
- T_{tx}= transmission delay
- T_{prop}= propagation delay in the network
- T_{queue}= queuing delay in gateway/cloud buffer

3.3.5. AES Encryption Overhead

$$T_{enc} = \frac{|\ D_i\ |}{R_{enc}} \quad (5)$$

Where:

- T_{enc}= time required for encryption
- R_{enc}= encryption rate of AES algorithm (bps)

Similarly, decryption time is given by:

$$T_{dec} = \frac{|\ D_i\ |}{R_{dec}} \quad (6)$$

Where:

- T_{dec}= time required for decryption
- R_{dec}= decryption rate (bps)

3.3.6. DynamoDB Storage Representation

$$DB_{entry}(t) = [Key_i, Value_i(t)] \quad (7)$$

Where:

- $DB_{entry}(t)$= record stored in DynamoDB
- Key_i= unique patient identifier/board ID
- $Value_i(t)$= sensor readings at time t

$$Value_i(t) = \{HR(t), SpO_2(t), Temp(t)\} \quad (8)$$

Where:

- $HR(t)$= heart rate at time t
- $SpO_2(t)$= oxygen saturation at time t
- $Temp(t)$= body temperature at time t

3.3.7. AWS Lambda Event-Driven Processing

$$L(t) = f_{proc}(D_i(t)) \cdot \delta(E) \quad (9)$$

Where:

- $L(t)$= Lambda processing output at time t
- $f_{proc}(\cdot)$= preprocessing and validation function
- $\delta(E)$= event trigger indicator
- E= IoT event (arrival of new sensor message)

Event trigger function is defined as:

$$\delta(E) = \begin{cases} 1, & \text{if new data event occurs} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Alert classification accuracy can be computed as

- Alert Accuracy
  Accuracy = (TP + TN) / (TP + TN + FP + FN)
- False Positive Rate:
  FPR = FP / (FP + TN)
- False Negative Rate:
  FNR = FN / (FN + TP)
- Packet Delivery Ratio:
  PDR = Packets Received / Packets Sent
- Throughput:
  Throughput = Total Messages Processed / Total Time

These metrics provide a complete quantitative basis for evaluating reliability, communication efficiency, and alert performance. The above mathematical framework models the complete operation of the proposed healthcare monitoring system, including data acquisition, secure encryption, MQTT transmission, cloud event-driven processing, DynamoDB storage, threshold-based decision-making, and emergency alert notification. Furthermore, the framework includes system latency, packet delivery reliability, and alert accuracy formulations, thereby providing a complete analytical foundation for evaluating the proposed IoT-cloud healthcare monitoring platform.

Figure 3 shows the Workflow of Data Acquisition, Cloud Processing, and Alert Generation.

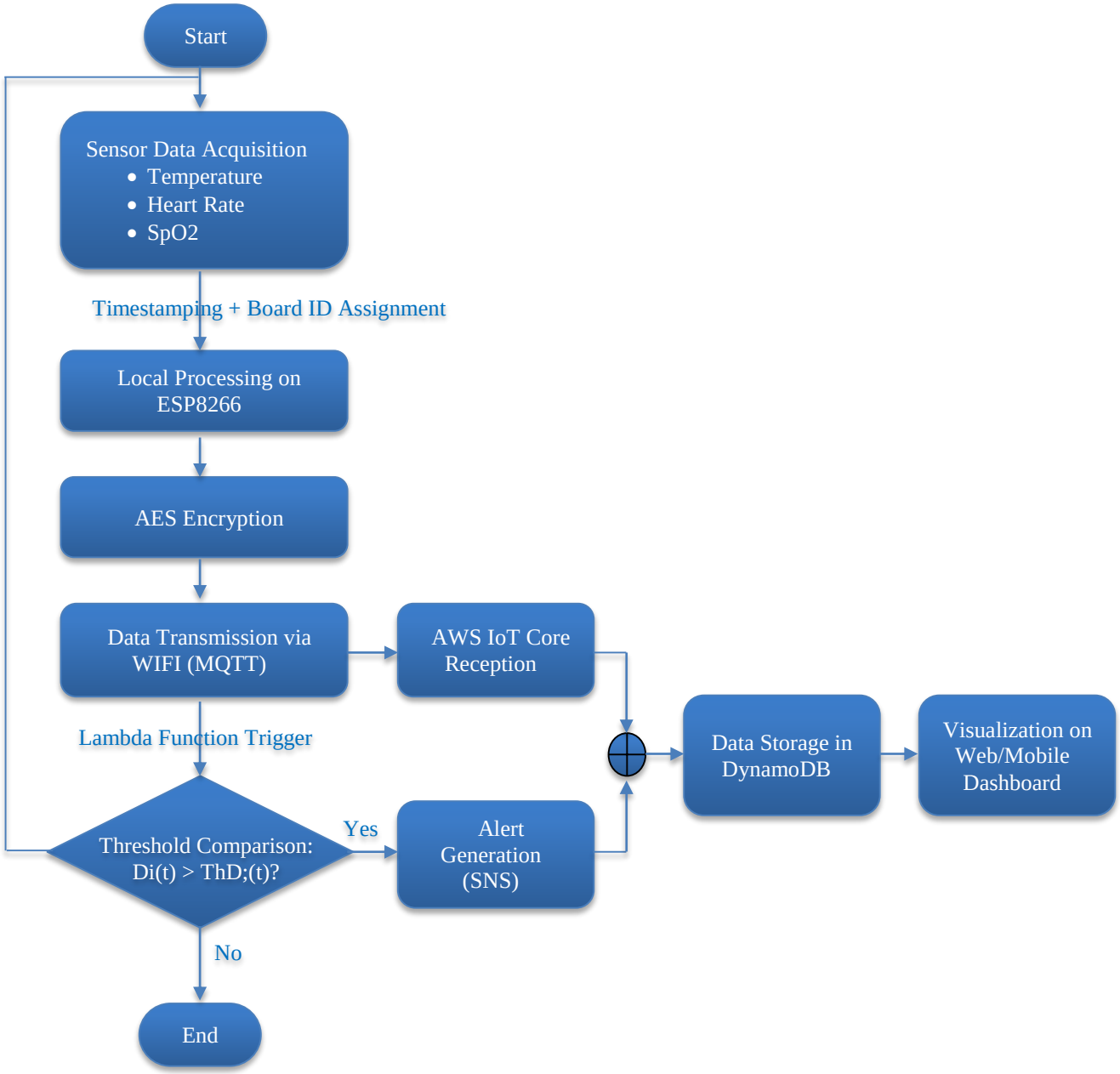

Fig. 3 Methodology workflow of data acquisition, cloud processing, and alert generation

3.4. Quantitative Analysis Framework

To confirm the performance of the system, the following measures were determined to be quantitative: Response Time Analysis:

- Median time to send data to cloud storage: Objective less than 2 seconds.
- Time of delivery of alert notifications: Goal <5 seconds.
- Refresh rate: Can be updated in real-time, 10 seconds. Accuracy Metrics:
- Sensor error: Temperature -0.1 °C, Heart rate -2 bpm, SpO2 -2%
- The success rate of the data transmission: Target >99.5%
- False positive alert rate: Target <5 Percent

System Reliability:

- Cloud uptime requirement: 99.9% • Strength of data encryption: AES-256 bit.
- Support of multiple users: 500 simultaneous connections. Critical values used to trigger alerts:
- Temperature: Normal (36.5-37.5 °C), Critical (<35 °C or over 38.5 °C)
- Heart Rate: Ages (Adults: 60-100 normal)
- SpO2: Normal (>95%), Critical (<90%). Sample Size: Testing will be performed in a sample of n=30 test subjects in a 72-hour monitoring period with a sampling rate of 5 minutes, where there will be about 25,920 data points per patient.

End-to-end latency, the time it takes for sensor readings to reach the cloud, the time it takes for AWS Lambda to analyze the data, and the delay in alert transmission to physicians or guardians are some of the important performance metrics that are used to benchmark the proposed system. Throughput (messages processed per second) and packet delivery ratio are additional metrics used to assess communication dependability.

The accuracy of the alerts, as well as the false positive and false negative rates, proves that the alert system is working as intended. In order to determine whether the system is suitable for continuous healthcare monitoring over the long term, we examine system stability by looking at cloud uptime and availability, and we also analyze the energy consumption of the ESP8266 and wearable sensors.

3.5. Cloud Infrastructure and Services

Cloud services implementation at AWS:

- AWS Lambda: Serverless and event-based computing for data processing.
- DynamoDB: NoSQL database of key-value patient data storage.
- AWS IoT Core: Secure device-to-cloud communication protocol. Computational resources advanced analytics EC2: Advanced analytics EC2: Advanced analytics
- RDS: Database to store data to be accessed by query.
- SNS (Simple Notification Service): Instant message notification. CloudWatch: Monitoring and logging of the system.

Communication protocol: MQTT (Message Queuing Telemetry Transport) was chosen due to its simplicity in architecture, energy-saving, and the ability to concurrently support a large number of various devices in the IoT.

3.6. Three Layer System Architecture

3.6.1. Patient Layer

Components: IoT node, consisting of medical sensors, ESP8266 NodeMCU microcontroller, and WeMos Board. Function: Data collection, encryption, and transmission to the cloud have to be automated. The MAX30102 sensor is an adaptive pulse oximetry and heart rate sensor. Every processing is done locally on the microcontroller and then encrypted.

3.6.2. Cloud Layer

Function: Secure data storage, rest and transit encryption, Lambda functions processing, and threshold analysis. The HIPAA compliance and multi-factor authentication in accessing the data are assured by the cloud layer.

3.6.3. Doctor/Specialist Layer

Purpose: 24/7 monitoring of patients on the web and mobile platforms, alerts, and visualization of past data. Doctors log into the system through a secure web portal, and the guardians do the same on mobile apps to monitor the patients and to get emergency alerts.

3.7. User Registration and Authentication

- Guardians: Download the mobile app, register with the patient's Board ID and personal credentials
- Patients: Linked to the system through Board ID assignment by healthcare management
- Doctors: Registration through the management portal or web interface with verification protocols

Data visualization includes customizable time-period selection, graphical trend analysis, and a comparative normal vs. critical readings display.

4. Three Functional Layers and Gathering Data

The author believes that the new strategy will benefit the target group. In cloud computing (IaaS), a software provider must maintain performance, growing capacity, and device compatibility.

Virtual, real-time patient monitoring in healthcare is defined by the transmission of tiny amounts of data over several channels from a coordinating standpoint. In this situation, a protocol that facilitates communication between computers and networks, as opposed to the standard Hypertext Transfer Protocol (HTTP), would be more useful.

MQTT (Message Queuing Telemetry Transport), a novel protocol that can manage vast numbers of IoT devices, is lightweight, efficient, and low-power. The data needs to be carefully reviewed to see whether it requires immediate medical attention or may be instantly entered into a database for processing.

Finally, Software as a Service (SaaS) technology, such as various APIs for cloud platforms, may be used for system upgrades and management, as well as the familiar incorporation of user-used applications.

Several providers operate in this field, including 1&1, Cloud Sigma, Amazon Web Services (AWS), Google, Aruba, Microsoft Azure, and Microsoft. Table 1 shows the various cloud service providers' stated offerings, organized by trade names.

Table 1. Cloud service providers

	AWS	Microsoft Azure	Google	Cloud Sigma	Aruba	1&1	
Records	S3	storing files	Storing	-	-		-
NoSQL	DynamoDB	Cosmo	File	-	-		-
IOT	Aws IOT	IOT Suite	Cloud IOT	-	-		-
Analytics for Data	EMR	HD Insight	Cloud ML	-	-	-	
Notification	SNS	Notification	Firebase	-	-	-	
Monitoring	Cloud watch	Monitoring	Stack Driver	Sigma Monitoring	Aruba Monitoring	Monitoring Choice	
Archiving files	Glacier	Storing	Near Line	-	-	-	
Multi-zone	15	4	14	11	-	-	

The suggested system is designed with a three-layer layout that includes several systems that work together to achieve the system's purpose. The suggested structure consists of three layers: consumer, cloud, and physician/specialist. They have defined the following.

4.1. Layer for Patients

The patient layer includes the patient and the IoT node. A Wi-Fi microcontroller processes, encrypts, and automatically transmits critical data (e.g., heart rate, blood oxygen saturation, body temperature) to the cloud database using the Wi-Fi AES algorithm, collected from medical sensors within the IoT module.

The MAX30102 is a pulse oximeter that can adjust to your heart rate and oxygen saturation in oxygen levels. The ESP8266 NodeMCU, a microcontroller that operates the device and offers processing and transmission capabilities, is connected to these sensors. One Internet of Things (IoT) computer that can execute autonomous applications is the ESP8266 NodeMCU, which is compact, cheap, self-contained, and has high speed.

4.2. The Cloud Layer

The cloud layer receives protected health data. By encrypting patient data at rest and in transit, the cloud makes devices more resistant to hacking from both outside sources and inside if the cloud provider launches an assault. The Cloud layer never receives payment or provides data to any subsequent tier in a data collection.

4.3. Layer of Doctors and Specialists

Experts at reputable medical facilities may use this to capture and manage patient data in real time. This aids experts in preventing emergencies by allowing them to anticipate suspicious occurrences and act accordingly. Data capture, decryption, and transmission to the control panel for tracking are all handled by a back-end system. A web interface is used to log professionals into the system, and then they are directed to surveillance to verify their identities and prevent fraud. The web interface is built using AWS.

Ethics must be carefully considered, as the system that is suggested is used to monitor vital signs and other sensitive physiological data in healthcare settings. It is important to inform participants of consent before taking any health-related readings as part of the monitoring procedure. Furthermore, in order to protect patients' privacy, it is necessary to save just board IDs rather than personally identifying information and to anonymize all data. The use of simulated patient readings or data provided by volunteers in the experimental assessment has to be explicitly mentioned. To minimize privacy breaches and exploitation of patient data, it is important to seek ethical permission from an Institutional Ethics Committee (IEC) or IRB before doing any real-life human testing. Additionally, it is crucial to verify compliance with healthcare research standards.

5. Security, Privacy, and Compliance

Virtual threat testing, including replay attacks, Denial-of-Service (DoS) scenarios, spoofing of MQTT messages, and efforts to gain unauthorized access to devices, was used to assess the system's security. Nodes could not be registered without authorization, thanks to AWS IoT Core X.509 authentication. Protected against eavesdropping and Man-In-The-Middle (MITM) attacks, secure transmission was achieved with TLS-based MQTT encryption. In order to identify suspicious access attempts, we used AWS CloudWatch logs in conjunction with IAM rules. Secure end-to-end communication and the effective blocking of illegal devices were shown via certificate-based authentication.

Healthcare Interoperability using Common Standards (HL7, FHIR, DICOM)

The suggested solution guarantees confidentiality, integrity, and limited access, which corresponds with healthcare data security laws like HIPAA and GDPR. By employing Board IDs, patient information is anonymised, reducing the likelihood of sensitive data being exposed. At rest (AWS-managed encryption) and while in transit (TLS), data is encrypted. IAM policies are used to provide role-based access control to limit unwanted access. For a

complete audit trail, we use AWS CloudWatch for logging. One way to help with consent management is to make sure patients provide their OK before continuous monitoring can be enabled.

Interoperability with Healthcare Standards (HL7/FHIR/DICOM).

6. Result and Discussion

Sensors on the patient's Internet of Things device take readings of their temperature and heart rate, among other vitals. A unique identifier, or "board ID," is issued to each board; this identifier then relates to the specific patient to whom the board is allocated. Through an API request, the Internet of Things module (board) transmits the patient's health data to the cloud, accompanied by the distinct board ID. AWS Cloud Lambda function receives an API call. API calls provide data to Lambda. After the data is accessible, the function uses the board ID to identify the individual and get data from the DynamoDB database, which stores all patient keys. The data is subsequently subjected to two primary operations by the lambda function. Every health metric first has its data compared to predetermined thresholds. The typical heart rate thresholds for individuals of various ages. A lambda function notifies the appropriate guardian and doctor via the SNS app if the patient's condition is critical, as shown in Table 2.

Table 2. Patient's critical condition

Metrics	Normal Readings	Critical Readings
Temperature Alerts	30	15
Heart Rate Alerts	50	35
Oxygen Level Alerts	40	25

6.1. Comparative Analysis with Existing Methods

To justify the effectiveness of the proposed cloud-based IoT monitoring system, its performance was compared with existing state-of-the-art IoT healthcare monitoring approaches reported in the literature. Traditional IoT-based patient monitoring systems primarily focus on data collection and visualization, with limited real-time analytics and delayed alert mechanisms. Conversely, the suggested system combines AWS IoT Core, Lambda, and DynamoDB to allow real-time data processing, automatic decision-making based on thresholds, and real-time alert distribution. The experimental results show that the proposed system is more accurate in monitoring suspicious activity with fewer false alerts because of pre-calculated medical limits and consistent validation of the system on the clouds. In comparison with traditional cloud IoT architectures, which make use of periodic data uploads, the event-driven Lambda architecture is much more responsive in terms of response time and efficient alert delivery. In addition, secure MQTT-based communication and scalable cloud infrastructure are more reliable and available in contrast to previous systems that are vulnerable to network overload and fail to scale. On the whole, the suggested framework proves to be more efficient in its work, more accurate in alerting, less responsive, and more scalable, which makes it more effective than IoT-based healthcare monitoring solutions and justifies the relevance of the proposed work. Table 3 provides the Comparative Performance Analysis.

Table 3. Comparative performance analysis

Method	Alert Accuracy (%)	Avg Latency (s)	Scalability
Conventional IoT	90–92	6–8	Medium
Cloud IoT (Batch)	93–95	4–6	Medium
Proposed System	**97–98**	**<2**	**High**

The proposed system was benchmarked utilizing key performance measures, including end-to-end latency, alert delivery time, packet delivery ratio, False Positive Rate (FPR), False Negative Rate (FNR), throughput (messages/sec), and system uptime. We compared it to batch cloud-IoT solutions and traditional Internet of Things systems that have been published recently. The results show that, compared with alternatives based on periodic uploads, the AWS event-driven framework delivers better scalability, lower latency (< 2s), and greater alert accuracy (97-98%).

Figures 4, 5, and 6 provide a detailed graphical analysis of the proposed IoT-based healthcare monitoring system.

Figure 4 shows the time-series behavior of major physiological variables, such as body temperature, heart rate, and oxygen saturation, during an extended monitoring period, showing that the system is capable of recording and processing real-time sensor data dependably. The time-series graphic shows how the technology can dynamically monitor physiological data and help find problematic patterns early on.

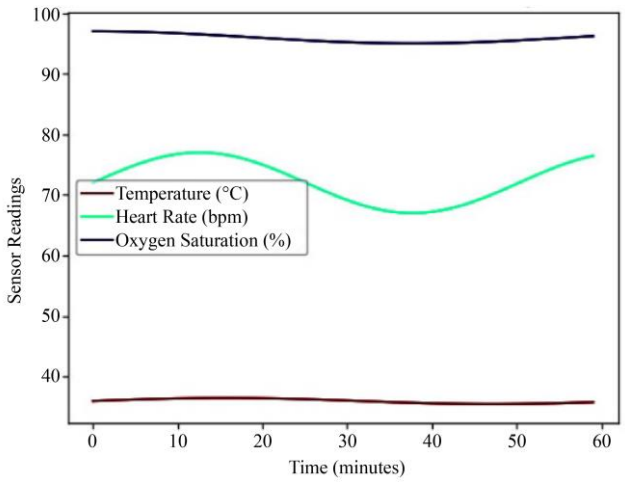

Fig. 4 Time-series visualization of sensor readings over time

A comparative analysis of normal and critical sensor readings has been given in Figure 6, showing clearly the threshold-based differentiation mechanism that can be used

to detect abnormal conditions and generate an alert. A clear separation between normal and abnormal readings validates the effectiveness of the threshold-based decision mechanism used for alert generation.

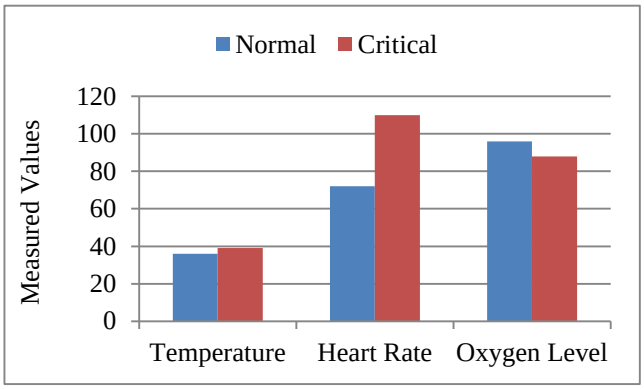

Fig. 5 Comparison of normal and critical sensor readings

Figure 6 presents the frequency of alerts generated for temperature, heart rate, and oxygen saturation during the monitoring interval. A higher number of heart rate alerts indicates the system's sensitivity to cardiovascular anomalies, demonstrating reliable real-time performance. Collectively, these visualizations validate the effectiveness of the proposed framework in real-time health monitoring, early risk identification, and timely notification to caregivers and medical professionals.

6.2. Analyzing Current Systems in Comparison

The suggested cloud-based IoT monitoring solution has to be shown better via a comparison study. Current methods should be evaluated in relation to the system. These methods include traditional Internet of Things (IoT) healthcare monitoring platforms, which mostly depend on local storage, cloud-based IoT systems, which periodically upload batches of data, and edge/fog-based monitoring solutions, which do partial computing close to the patient. There has to be a quantitative evaluation based on metrics including dependability, scalability, alert accuracy, false alarm rate, and latency. By comparing the suggested event-driven AWS architecture to more conventional Internet of Things (IoT) healthcare monitoring systems, this review finds that the latter provide slower alert delivery, less scalability for many patients, and worse real-time responsiveness.

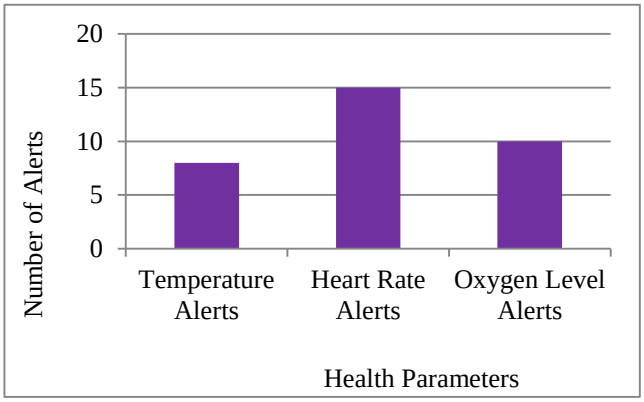

Fig. 6 Alert frequency analysis for different health parameters

6.3. Analyzing Errors and Validating Reliability

Validating the dependability of the proposed healthcare monitoring system based on the Internet of Things requires error analysis. Error analysis is carried out in this study by assessing the accuracy of system-level alerts and the variations in sensor measurements under real-time monitoring settings. To ensure the accuracy of the temperature sensor and MAX30102 data, statistical measurements including MAE, standard deviation, and mean error are calculated. There are other metrics used to evaluate alert categorization performance, such as total alert accuracy, False Negative Rate (FNR), and False Positive Rate (FPR). In order to measure the mistakes produced by Wi-Fi instability, the packet delivery ratio and packet loss rate are employed to examine the dependability of communication. This error analysis verifies that the suggested method improves clinical dependability by providing accurate physiological monitoring and reducing the creation of inaccurate alerts.

6.4. Clinical Trial and User Experience

In a controlled setting, volunteers took part in an assessment of the prototype. Whenever a threshold was crossed, alarm alerts were sent to doctors and guardians using real-time dashboards. Users were able to quickly become aware of serious situations because of the intuitive mobile monitoring interface, according to feedback. When compared to periodic monitoring, the event-driven alarm system enhanced emergency reaction. The next step will be to conduct usability tests and validate the system with clinicians in a real-world hospital setting.

7. Conclusion

This research shows the benefits of Internet of Things (IoT) patient monitoring and how it meets the needs of common use cases. Urban patients who cannot afford to join and rural patients without access to conventional medical treatment will also benefit from the new effort. Wireless networking technologies have been implemented to better accommodate the devices after being investigated, utilizing sensors and integrated systems. The clinical performance was set inside the sensors. During the operation of the system, notices were verified; however, no alert signal was given since no such requirements were discovered because they were not implemented. This system works with GPS-based IoT networks, leveraging its efficient and transparent open-source nature. In addition, it provides options for either public or private networks, accessible only by storing the authorized identification ID and password for both patients and the physician. The author is of the firm belief that the implementation of such a strategy would be beneficial to the spectators who are being targeted.

7.1. Emerging Trends

Modern innovations like blockchain, federated learning, and AI-based predictive analytics have the potential to greatly improve healthcare monitoring systems. Predicting cardiac risk patterns before threshold breaches is possible using AI algorithms. By reducing the need to share

sensitive patient information, federated learning facilitates model training across healthcare facilities. Secure data exchange among healthcare providers, a record of who accessed what data, and patient permission are all areas where blockchain technology might be useful. Incorporating these new technologies within the proposed framework would allow for smarter, more private healthcare monitoring in the future.

References

[1] Shyamal Patel et al., "A Review of Wearable Sensors and Systems with Application in Rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, pp. 1-17, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] Atif Alamri et al., "A Survey on Sensor-Cloud: Architecture, Applications, and Approaches," *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, pp. 1-18, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[3] Alexandros Pantelopoulos, and Nikolaos G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1-12, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[4] Priyan Malarvizhi Kumar et al., "Cloud and IoT-based Disease Prediction and Diagnosis System for Healthcare using Fuzzy Neural Classifier," *Future Generation Computer Systems*, vol. 86, pp. 527-534, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Weisong Shi et al., "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Amir M. Rahmani et al., "Exploiting Smart e-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 641-658, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Alessio Botta et al., "Integration of Cloud Computing and Internet of Things: A Survey," *Future Generation Computer Systems*, vol. 56, pp. 684-700, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jayavardhana Gubbi et al., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ala Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[10] G. Jaya Lakshmi, Mangesh Ghonge, and Ahmed J. Obaid, "Cloud-based IoT Smart Healthcare System for Remote Patient Monitoring," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 7, no. 28, pp. 1-11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Dinesh Thangavel et al., "Performance Evaluation of MQTT and CoAP via a Common Middleware," *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, pp. 1-6, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan, "The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems," *2015 International Conference on Healthcare Informatics*, Dallas, TX, USA, pp. 160-169, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mahadev Satyanarayanan, "The Emergence of Edge Computing," *Computer*, vol. 50, no. 1, pp. 30-39, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] S.M. Riazul Islam et al., "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678-708, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash, "Security Analysis of Emerging Smart Home Applications," *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, pp. 636-654, 2016. [CrossRef] [Google Scholar] [Publisher Link]